In the Claims:

Claim 1 (Currently Amended)

1. For use in a data processing system having a memory coupled to multiple requesters, a memory coherency system, comprising:

a memory circuit coupled to provide a copy of requested data from the memory to a first requester, and to initiate invalidation operations to invalidate all read-only copies of the requested data that are stored by one or more other requesters, wherein the data is provided before the invalidation operations are completed; and

a circuit included within the first requester and responsively coupled to the memory circuit to execute an instruction that causes the first requester to temporarily enter a stalled state until all of the invalidation operations have been completed.

Claim 2 (Cancelled)

Claim 3 (Original) 1

3. The system of Claim 2, wherein the memory circuit includes a request channel and a response channel.

Claim 4 (Original) 2

4. The system of Claim 3, wherein the memory circuit includes an acknowledge tracker to initiate the transfer of an acknowledge to the first requester when all invalidation operations for the requested data are completed.

Claim 5 (Original)

4 ~~5~~. The system of Claim ~~4~~ 3, wherein the requester includes a request tracking circuit responsively coupled to the memory to record when the acknowledge is outstanding for the requested data.

Claim 6 (Original)

5 ~~6~~. The system of Claim 1, wherein the first requester issues multiple requests, and wherein the circuit prevents any further instruction processing from occurring within the first requester until all invalidation operations have been completed for all of the multiple requests.

Claim 7 (Original)

6 ~~7~~. The system of Claim ~~6~~ 5, wherein the first requester is a processing node that includes multiple processors, wherein the circuit resides within one of the multiple processors and includes logic to execute an instruction to stall the processor until all of the invalidation operations have been completed for data previously provided to the processor.

Claim 8 (Original)

7 ~~8~~. The system of Claim ~~6~~ 5, wherein the first requester is a processing node that includes multiple processors, wherein the circuit resides within one of the multiple processors and includes logic to execute an instruction to stall the processor until all of the invalidation operations have been completed for data previously provided to predetermined ones of the processor in the processing node.

Claim 9 (Original)

8 ~~9~~. For use in a system having multiple requesters coupled to a shared memory, a method for controlling processing of requests, comprising:

a.) issuing a request for data by a requester to the shared memory;

b.) providing the data from the shared memory in response to the request before all read-only copies of the data retained by other requesters have been invalidated; and

c.) stalling the requester until all of the read-only copies have been invalidated.

Claim 10 (Original) 9 8

~~10.~~ The method of Claim ~~9~~, wherein step c.) includes the initiation by the requester of a hardware sequence to stall the requester until the read-only copies have been invalidated.

Claim 11 (Original) 10 9

~~11.~~ The method of Claim ~~10~~, wherein the requester is an instruction processor, and further including execution of a predetermined instruction to initiate the hardware sequence.

Claim 12 (Original) 11 10

~~12.~~ The method of Claim ~~11~~, wherein the instruction is part of the hardware instruction set of the instruction processor.

Claim 13 (Original) 12 8

~~13.~~ The method of Claim ~~9~~, and including repeating steps a.) and b.) for multiple requests, and stalling the requester until all read-only copies of any data requested by any of the multiple requests have been invalidated.

Claim 14 (Original) 13 8

~~14.~~ The method of Claim ~~9~~, wherein the requester is a processing node containing multiple processors, and wherein the method comprises:

a.) issuing a request for data by one of the processors to the shared memory;

b.) providing the data from the shared memory to the processor in response to the request before all read-only copies of the data retained by other requesters have been invalidated; and

c.) stalling the processor until all of the read-only copies have been invalidated.

Claim 15 (Original) 13

14 ~~15~~. The method of Claim ~~14~~, wherein a.) and b.) are repeated for multiple requests, and wherein step c.) includes stalling the processor until all read-only copies of any data previously provided to the processor have been invalidated.

Claim 16 (Original) 14

15 ~~16~~. The method of Claim ~~15~~, wherein steps a.) through c.) may be performed for more than one processor in the processing node, and wherein step c.) comprises stalling a processor until all read-only copies of any data previously provided to the processor have been invalidated.

Claim 17 (Original) 14

16 ~~17~~. The method of Claim ~~16~~, wherein steps a.) through c.) may be performed for more than one processor in the processing node, and wherein step c.) comprises stalling a processor until all read-only copies of any data previously provided to predetermined ones of the processors in the processing node have been invalidated.

Claim 18 (Original)

17 ~~18~~. The method of Claim 9, and further comprising issuing an inter-processor interrupt by the requester to another requester to indicate that data stored within the shared memory by the requester may be accessed by the other requester.

Claims 19-22 (Canceled)